Stacked Ensemble Model for Predicting Fatal Myocardial Infarction Complications

Rophence Ojiambo

Introduction

Cardiovascular diseases (CVDs) remain the leading cause of morbidity and mortality worldwide. In 2019 alone, CVDs accounted for approximately 17.9 million deaths, representing 32% of all global deaths, with Myocardial infarction (MI) affecting over 7 million individuals globally each year[1–3]. Additionally, MI is marked by irreversible cardiac muscle necrosis due to prolonged ischemia, and is a leading cause of early in-hospital mortality, with acute complications such as ventricular arrhythmia, cardiogenic shock, pulmonary edema, and myocardial rupture driving most deaths within 72 hours of admission[4,5]. Timely identification of high-risk patients is critical to enable early clinical intervention, reduce mortality, and allocate limited healthcare resources effectively, making accurate prediction tools essential from a public health perspective. Machine learning (ML) enables predictive modeling by extracting patterns from clinical data, including demographics, biomarkers, and treatments[6]. However, standalone ML models such as logistic regression, decision trees, or support vector machines often face limitations like overfitting or bias and reduced generalizability in heterogeneous datasets. Additionally, many predictive models lack interpretability or fail to leverage the complementary strengths of multiple algorithms, limitations that hinder their translation into routine clinical use. Ensemble methods, particularly stacking, mitigate these issues by integrating predictions from diverse algorithms (e.g., bagging, boosting) through a meta-learner, optimizing accuracy and robustness[7,8]. This approach leverages complementary strengths of base models, making it uniquely suited for predicting fatal MI complications where risk factors are multifactorial and interdependent. The growing availability of structured electronic health data presents an opportunity to develop more robust and scalable prediction models. However, there remains a need for approaches that can integrate heterogeneous features, address class imbalance, and provide transparent predictions suitable for clinical decision support. Ensemble learning, particularly stacking, has shown promise in improving predictive performance by combining diverse base learners through a meta-model, but its application to early MI complication prediction remains limited. To address these gaps, this study proposes a stacked ensemble machine learning framework for predicting early lethal complications of myocardial infarction using a publicly available clinical dataset of 1,700 patients. The approach includes feature selection using LASSO and recursive feature elimination, imputation and preprocessing of high-missingness variables, and model evaluation using discrimination and calibration metrics. This work aims to demonstrate the utility of ensemble modeling for high-risk patient identification and its potential for integration into public health bioinformatics tools for acute cardiovascular care.

Related Work

Deep Learning: Recent work has applied deep neural networks to MI prediction and multi-label outcome classification. For example, Abbas et al[9] employed various deep learning algorithms such as Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Deep Neural Network (DNN), Long Short-Term

Memory (LSTM) to detect MI and simultaneously classify multiple complication labels. Diakou et al[10] likewise explored multi-label classification on the same MI complications dataset, training models to predict all recorded complication categories (the dataset contains 1,700 patients with 114 features and 12 outcome labels). These deep-learning studies reported high accuracy in identifying MI outcomes, but (like many neural models) they offer limited interpretability of how predictions are made. In particular, complex models can be "black boxes," making it hard to trace which clinical features drive a given prediction. This motivates our focus on interpretability alongside predictive performance.

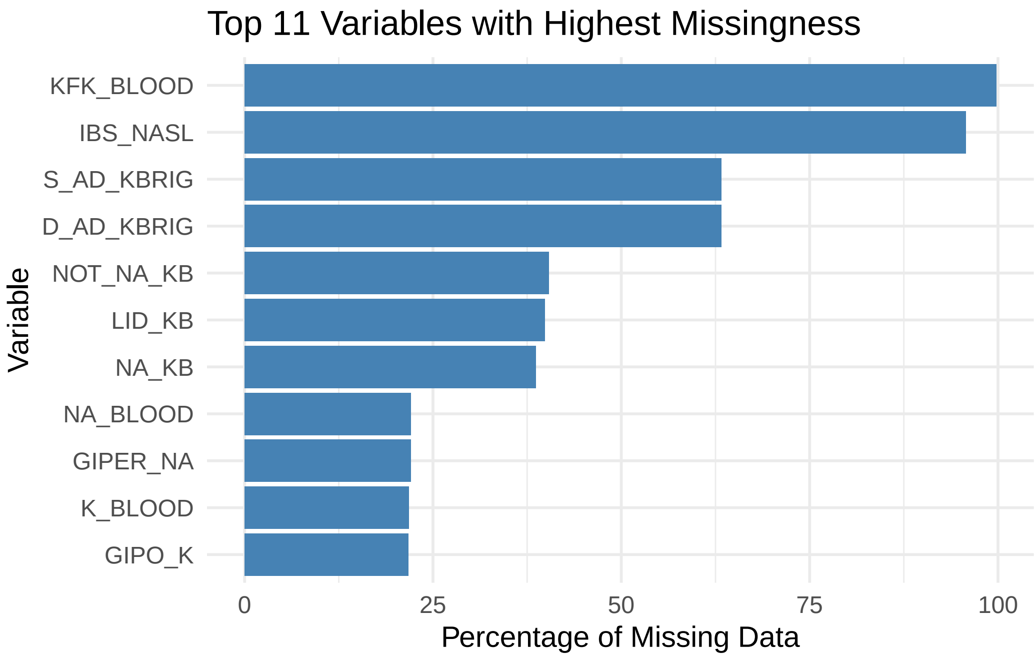
Classical Machine Learning: Other studies have used conventional machine learning algorithms on the UCI MI complications data. Satty et al[11] performed a comparative evaluation of several classifiers; Multilayer Perceptron (MLP), Naive Bayes (NB), and Decision Tree (DT) to predict lethal MI outcomes. They applied these models both on the full feature set and on a reduced set obtained by WEKA (Waikato Environment for Knowledge Analysis) machine learning software attribute selection. Results showed that the MLP achieved the best accuracy on the full dataset, while the decision tree performed best after feature reduction. Similarly, Ghafari et al[12] tested four common classifiers; logistic regression, support vector machine, random forest, and extreme gradient boosting (XGBoost) with recursive feature elimination on the same data. They reported that XGBoost outperformed the others, achieving an AUC of about 78.7% and accuracy around 91.5%, with cardiogenic shock emerging as the top predictive feature. These classical ML studies demonstrate that simple models can achieve reasonable predictive power on MI outcomes. However, they generally focus on single algorithms and accuracy metrics, without leveraging ensemble methods or explaining model behavior. In each case the task was treated as a binary classification (fatal complication vs. survival), rather than a richer multi-label problem.

Ensemble Methods: In contrast to the above single-model approaches, stacked or ensemble methods can further boost performance by combining multiple classifiers. Ensembles have proven effective in other cardiovascular prediction tasks (often improving robustness and accuracy), yet they have been underexplored on the UCI MI dataset. Moreover, recent literature highlights the importance of model interpretability in clinical settings: for instance, Tarabanis et al[13] emphasize that "lack of explainability" in ML models hinders clinical adoption, and they explicitly develop explainable XGBoost models for MI mortality prediction.

Limitations of Prior Work: Overall, existing MI prediction studies have certain limitations. Most rely on a single data modality (e.g. tabular clinical records or ECG features) and on simpler models. The deep-learning work[9,10] achieves high accuracy but does not address model explainability. The classical ML studies[11,12] demonstrate which algorithms perform well, but do not leverage modern ensemble techniques or provide detailed insights into feature importance. By contrast, this project directly targets these gaps: we apply a stacked ensemble (expected to exceed single-model performance) and generate interpretable explanations, bridging the methodological advances of recent literature while addressing their shortcomings.

Data and Experiment Setup

The dataset used in this study, the Myocardial Infarction Complications database, contains detailed clinical records from 1,700 patients admitted to the Krasnoyarsk Interdistrict Clinical Hospital (Russia) between 1992 and 1995. It is publicly available through the UCI Machine Learning Repository.. The dataset comprises 111 predictors spanning **patient demographics** (e.g., age, sex, obesity), **cardiovascular history** (e.g., prior MI, hypertension stage), **ECG findings** (e.g., arrhythmias, infarct localization), **lab results** (e.g., serum potassium, sodium, WBC count, ESR), and **treatment variables** (e.g., opioids, NSAIDs, fibrinolytics).The dataset exhibited moderate overall missingness (7.6%), although some features had greater than 50% missingness. The bar plot below illustrates the top 11 variables in the dataset with the highest percentages of missing data. The variable `KFK_BLOOD` exhibits the most significant missingness, with nearly 100% of its values absent, followed closely by `IBS_NASL`, which also approaches complete missingness.



Variables such as S_AD_KBRIG and D_AD_KBRIG have around 75% missing data, indicating considerable data sparsity. Mid-range missingness is observed in variables like NOT_NA_KB, LID_KB, and NA_KB, each with approximately 50% of their values missing. On the lower end of the top 11, variables including NA_BLOOD, GIPER_NA, K_BLOOD, and GIPO_K have missingness levels closer to 25%.

Data Preprocessing involved removing variables with >50% missingness (e.g., KFK_BLOOD, S_AD_KBRIG, D_AD_KBRIG), imputing continuous variables using the mean and categorical variables using the mode, and encoding ordinal and binary variables as labeled factors. A detailed data dictionary was constructed to document variable transformations and definitions, see project codebook. We then proceed to clean the data by removing columns with excessive missingness, imputing missing values in continuous variables with the mean, and filling in nominal and ordinal variables with the mode. We then converted multiple binary and ordinal variables into factors with meaningful labels to improve interpretability. Among the 12 recorded complication outcomes shown in the table below, we focused exclusively on the binary LET_IS variable, indicating lethal outcome within 72 hours. The outcome was heavily imbalanced, with 271 patients (15.94%) experiencing a fatal complication.creating challenges of class imbalance and high missingness.

Table 1: Complications Summary (Sorted by Prevalence)

Complication	# Cases	Fraction
Chronic heart failure	394	23.18%
Lethal outcome (cause)	271	15.94%
Atrial fibrillation	170	10.00%
Pulmonary edema	159	9.35%
Relapse of myocardial infarction	159	9.35%
Post-infarction angina	148	8.71%
Dressler syndrome	75	4.41%
Ventricular fibrillation	71	4.18%
Third-degree AV block	57	3.35%
Myocardial rupture	54	3.18%
Ventricular tachycardia	42	2.47%
Supraventricular tachycardia	20	1.18%

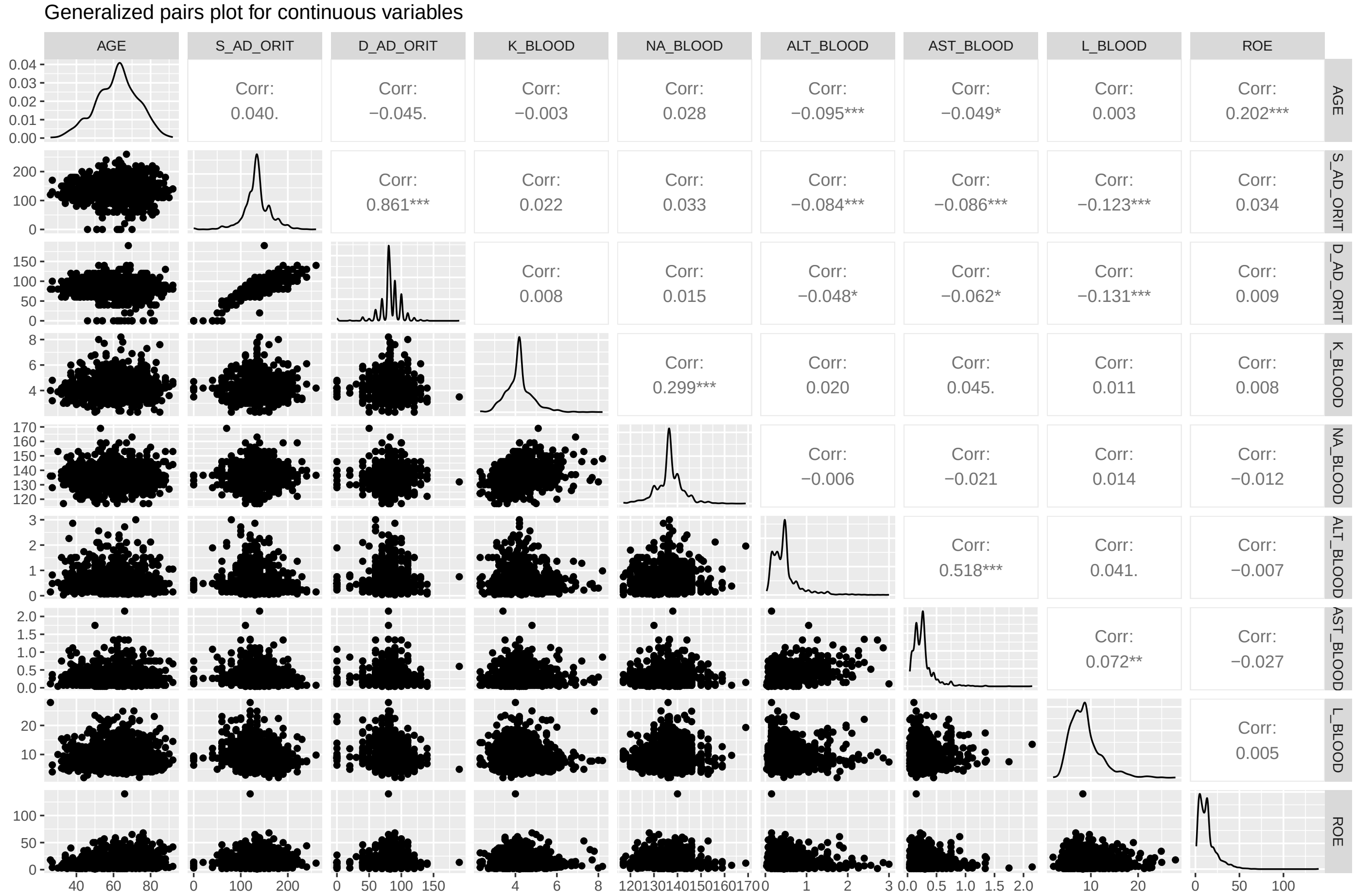


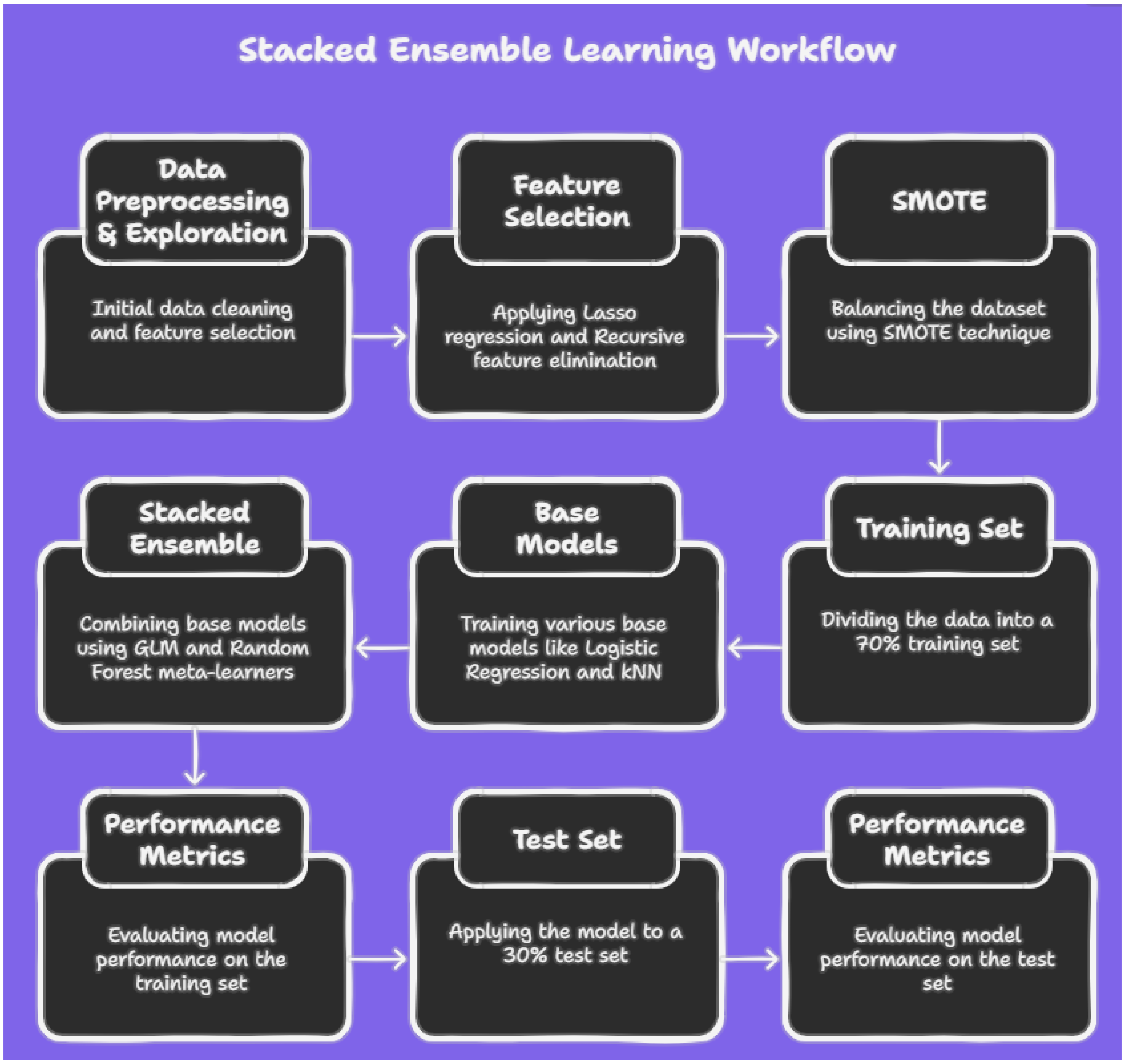
Table 2: Descriptive Statistics for Continuous Clinical Variables

	Min	Q1	Median	Mean	Q3	Max	SD
Age (years)	26.00	54.00	63.00	61.86	70.00	92.00	11.23
Systolic Blood Pressure ICU	0.00	120.00	134.59	134.59	150.00	260.00	28.78
Diastolic Blood Pressure ICU	0.00	80.00	82.75	82.75	90.00	190.00	16.82
Serum Potassium Content (mmol/L	2.30	3.80	4.19	4.19	4.50	8.20	0.67
Serum sodium content (mmol/L)	117.00	134.00	136.55	136.55	140.00	169.00	5.75
serum AIAT Content (IU/L)	0.03	0.23	0.45	0.48	0.52	3.00	0.35
Serum AsAT Content (IU/L)	0.04	0.15	0.23	0.26	0.30	2.15	0.18
White blood cell count (billion/L	2.00	6.60	8.40	8.78	10.20	27.90	3.27
Erythrocyte Sedimentation Rate (MM)	1.00	6.00	11.00	13.44	17.00	140.00	10.60

The pairs of scatter and density plots above provide an overview of the distributions for several clinical and biochemical variables. These variables exhibit diverse scales, units, and distribution shapes. For example, **Age**, **Systolic**, and **Diastolic Blood Pressure** show relatively symmetric or slightly skewed distributions, while **biochemical markers** such as **Serum Potassium**, **Sodium**, **ALT**, **AST**, and **Erythrocyte Sedimentation Rate (ESR)** are clearly right-skewed, with long tails extending toward higher values. Variables like **White Blood Cell Count** and **ESR** also display wide ranges, further emphasizing the heterogeneity in scale. This variation is supported by the summary statistics: age ranged from 26 to 92 years (mean = 61.9, SD = 11.2); systolic and diastolic blood pressure ranged from 0 to 260 mmHg and 0 to 190 mmHg, respectively (means = 134.6 and 82.8 mmHg). Serum potassium and sodium concentrations showed moderate variability (means = 4.19 and 136.6 mmol/L), while liver enzymes AIAT (ALT) and AsAT (AST) exhibited low median values but substantially high maximums, indicating potential outliers or acute elevations in a subset of patients. White blood cell count ranged from 2.0 to 27.9 billion/L (mean = 8.78), and ESR spanned from 1 to 140 mm/hr (mean = 13.4, SD = 10.6). These distributional patterns and descriptive characteristics in Table 2 highlight the clinical heterogeneity of the cohort and underscore the importance of normalization and careful preprocessing prior to modeling. Because of these differences in scale and distribution, **standardization** (e.g., z-score normalization) was essential before applying many machine learning algorithms, especially those sensitive to the magnitude of input features (like k-nearest neighbors, logistic regression, or neural networks). Standardizing the data helps ensure that variables contribute equally to distance calculations or model weights, improves convergence in optimization routines, and can reduce bias caused by dominant features with larger numeric ranges.

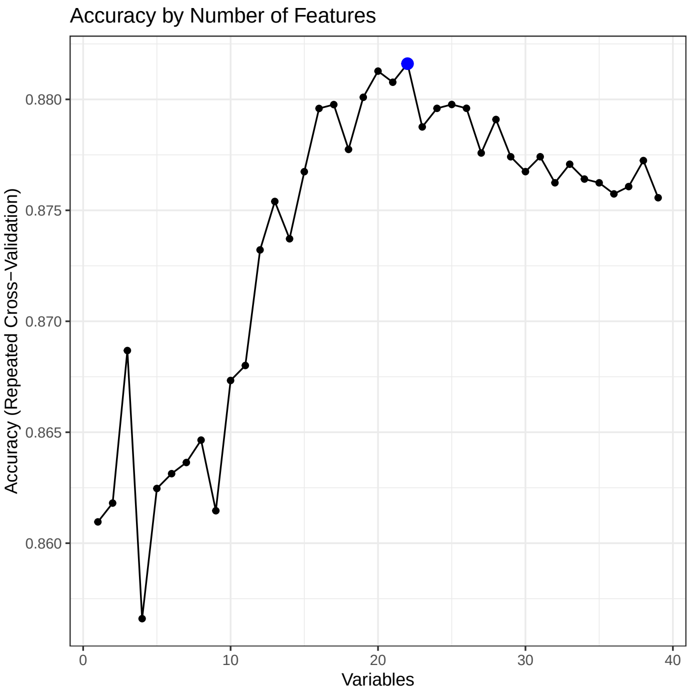
Methods

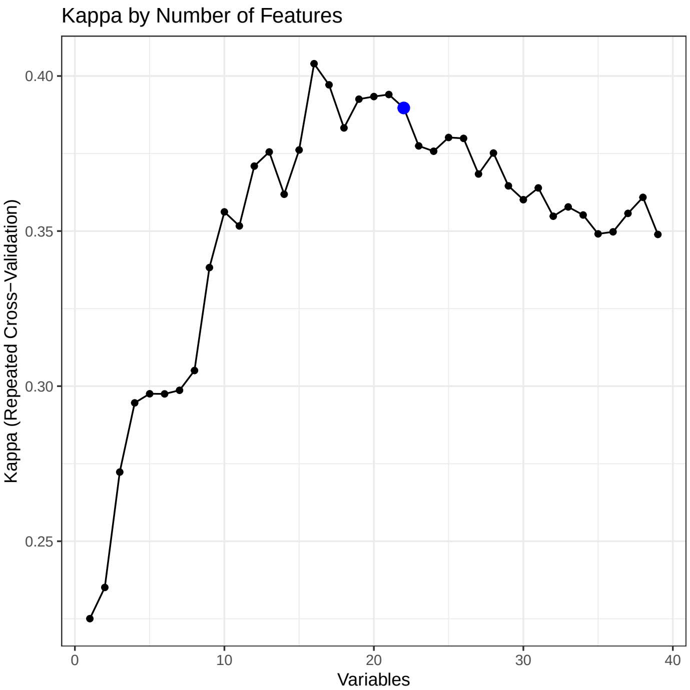
This study aimed to develop a predictive model for early in-hospital mortality following myocardial infarction (MI), defined by the binary outcome `LET_IS`, which indicates whether a patient died within 72 hours of admission. The analysis pipeline is illustrated in the figure below.


Stacked Ensemble Learning Workflow
Data Preprocessing & Exploration
Initial data cleaning and feature selection
Feature Selection
Applying Lasso regression and Recursive feature elimination
SMOTE
Balancing the dataset using SMOTE technique
Training Set
Dividing the data into a 70% training set
Base Models
Training various base models like Logistic Regression and kNN
Stacked Ensemble
Combining base models using GLM and Random Forest meta-learners
Performance Metrics
Evaluating model performance on the training set
Test Set
Applying the model to a 30% test set
Performance Metrics
Evaluating model performance on the test set

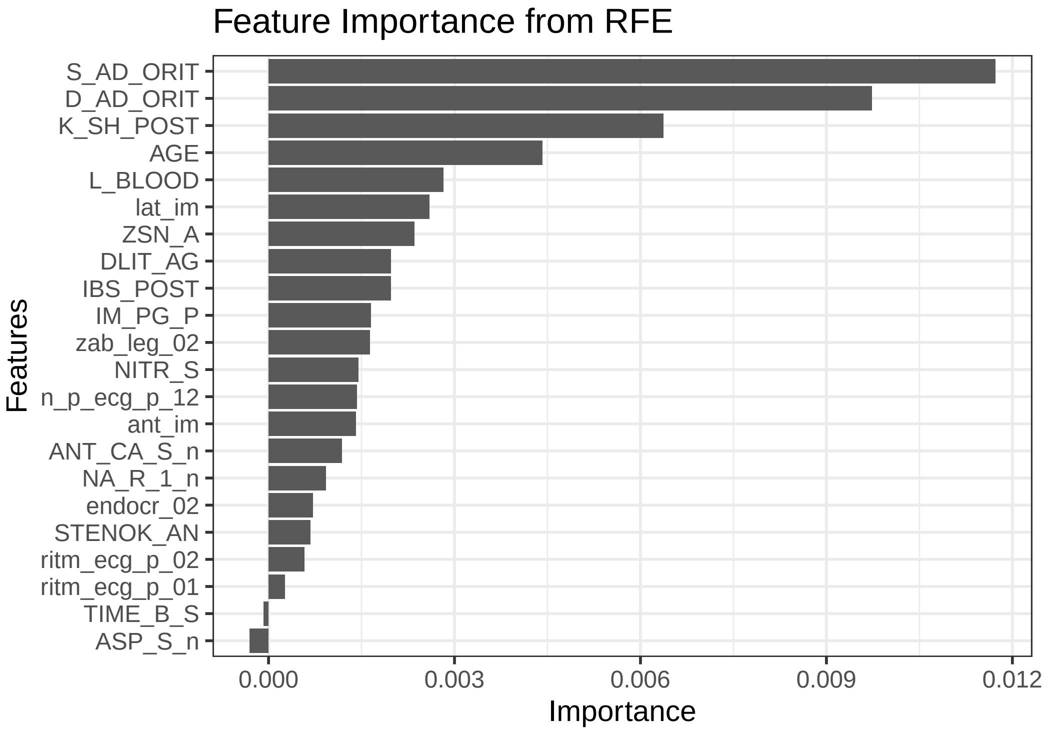
The objective was to classify patients as “Alive” or “Dead” based on 111 clinical, demographic, laboratory, ECG, and treatment variables drawn from the UCI Myocardial Infarction Complications dataset. Given the high-dimensional feature space, class imbalance (15.94% mortality), and data heterogeneity, we employed a stacked ensemble learning framework supported by data preprocessing and feature selection. Prior to feature selection, the dataset was randomly partitioned into training and testing subsets using `createDataPartition` to preserve the proportion of the binary outcome variable, `LET_IS`. 70% of the data was allocated to the training set, and the remaining 30% to the test set. Following the partitioning, only the relevant predictor variables (columns 1 to 107) and the outcome variable (column 119) were retained for analysis. Continuous variables in the training set were standardized using z-score normalization, where each variable was transformed to have a mean of zero and a standard deviation of one based on statistics calculated from the training data. This standardization was performed to ensure comparability across features and improve model performance.

To reduce dimensionality, prevent overfitting, and identify the most predictive features for the binary outcome `LET_IS`, a two-step feature selection process was employed. First, a **Least Absolute Shrinkage and Selection Operator (LASSO)** regression was conducted using the `glmnet` package. Predictor variables were extracted from the standardized training dataset using the model matrix, excluding the intercept. A 10-fold cross-validation procedure, with the area under the curve (AUC) as the performance metric, was used to determine the optimal penalty parameter (`lambda.min`). Variables with non-zero coefficients in the LASSO model were selected, yielding a set of 39 candidate predictors. Next, **recursive feature elimination (RFE)** was applied to this reduced feature set to further refine the selection. The RFE procedure was implemented using random forest functions (`rfFuncs`) and performed within a repeated 10-fold cross-validation framework (5 repeats). Parallel processing was employed to improve computational efficiency. The final RFE model identified the optimal subset of 22 features based on model performance across varying subset sizes. The results of the RFE procedure were saved for further evaluation.





As shown above, the first plots illustrate how model **Accuracy** and **Kappa statistics** varied as the number of variables increased. Both metrics peaked around 22 variables, as highlighted by the blue dot, suggesting this was the optimal number of predictors to balance performance and parsimony. The plot below presents Feature Importance for the selected variables, ordered by their contribution to the model. The final selected feature importance set consisted of the top 20 out of 22 predictors contributing most to model accuracy. This step ensures that only the most informative variables are retained for the final model, improving both interpretability and predictive performance.



The next step focused on using **SMOTE (Synthetic Minority Over-sampling Technique)** to address class imbalance in the outcome variable `LET_IS`. The process began by selecting the final features identified through RFE, and creating a cleaned training dataset (`train_final`) that included only these features and the outcome variable. Next, variable formats were harmonized in preparation for SMOTE: **Ordered categorical variables** were converted to numeric values to preserve their rank order, **Binary categorical variables** were transformed into 0/1 numeric indicators, the original factor levels were stored to allow for conversion back to interpretable labels after SMOTE. A **`recipe`** was defined to apply SMOTE (with an oversampling ratio of 0.7) and to drop any rows with missing values. After prepping and baking the recipe, the `train_balanced` dataset was generated, now with a more balanced distribution of the outcome (`Alive`: 59%, `Dead`: 41%). Finally, the transformed numeric variables were **converted back to their original factor levels** using the stored labels, ensuring data interpretability was retained post-balancing. A structural and summary check confirmed that the dataset was clean (no NAs) and ready for model training. Model training was performed using six **base classifiers:** `logistic regression (GLM)`, `k-nearest neighbors (kNN)`, `gradient boosting machine (GBM)`, `bagged decision trees`, `Random Forest (RF)`, and `eXtreme Gradient Boosting (XGBoost)`. Models were trained using repeated 10-fold cross-validation with hyperparameter tuning implemented through `caret::train`. Out-of-fold predictions were passed to **meta-learners**, `GLM` or `random forest` using the `caretStack` framework to construct 2 final stacked ensemble models. Performance was assessed on a held-out 30% stratified test set using `accuracy`, `sensitivity`, `specificity`, `precision`, `F1 score`, `false positive rate (FPR)`, `false negative rate (FNR)`, and `AUC`.

Several assumptions underpinned the modeling process: (1) Missing data were assumed to be **missing at random** (MAR), justifying mean/mode imputation and SMOTE application post-imputation.; (2) SMOTE-balanced training data were assumed to maintain generalizable patterns despite the use of synthetic minority examples; and (3) Predictors were assumed to have linear, additive contributions in base models like GLM, though ensemble learners relaxed this assumption through non-linearity. All analyses were implemented in R using the `tidyverse`, `caret`, `caretEnsemble`, `glmnet`, `xgboost`, and `themis` packages. Reproducibility was ensured through fixed seeds and `doRNG` for parallel processing.

Results

We evaluated six classifiers and two stacked ensemble models for predicting lethal complications within 72 hours using the MI Complications dataset. From the output results, we can see from the correlation Table 3 results that none of the individual ML algorithm predictions are highly correlated. Very highly correlated results mean that the algorithms have produced very similar predictions and combining similar predictions may not really yield significant benefit compared with what one would avail from accepting the individual predictions. In this specific case, we observe that none of the algorithm predictions were highly correlated so we straightforwardly moved to the next step of stacking the predictions.

Table 3: Correlation Matrix Between Base Learners

	GLM	**GBM**	**kNN**	**Bagged**	**XGB**	**RF**
GLM	1.000	0.120	-0.069	-0.136	0.088	-0.055
GBM	0.120	1.000	0.053	0.036	0.206	-0.006
kNN	-0.069	0.053	1.000	-0.046	0.007	0.020
Bagged	-0.136	0.036	-0.046	1.000	-0.049	-0.063
XGB	0.088	0.206	0.007	-0.049	1.000	-0.016
RF	-0.055	-0.006	0.020	-0.063	-0.016	1.000

As shown in Table 4, the stacked ensemble using a Random Forest meta-learner achieved the best overall performance on the training data, with an accuracy of **95.1%**, an F1 score of **0.959**, and an AUC of **0.984**. The stacked GLM followed closely with an accuracy of **94.6%** and an AUC of **0.987**. All ensemble models outperformed individual classifiers (e.g., GBM, XGBoost, RF) in terms of **F1 Score** and **Sensitivity**, indicating superior performance in detecting the minority class. Models such as *GLM*, *KNN*, and *GBM* had comparatively lower Sensitivity and F1 Scores.

Table 4: Model Performance Comparison in the Training Set

Model	**Accuracy**	**Sensitivity**	**Specificity**	**Precision**	**FPR**	**FNR**	**F1 Score**	**AUC**
GLM	0.8160	0.7457	0.8651	0.7945	0.1349	0.2543	0.7693	0.8991
GBM	0.8471	0.7900	0.8871	0.8303	0.1129	0.2100	0.8097	0.9264
KNN	0.8701	0.8671	0.8721	0.8258	0.1279	0.1329	0.8460	0.9474
BAGGED	0.9048	0.8900	0.9151	0.8799	0.0849	0.1100	0.8849	0.9761
XGB	0.9265	0.9286	0.9251	0.8966	0.0749	0.0714	0.9123	0.9832
RF	0.9306	0.9200	0.9381	0.9122	0.0619	0.0800	0.9161	0.9817
Stacked GLM	0.9462	0.9542	0.9349	0.9548	0.0651	0.0458	0.9543	0.9874
Stacked RF	0.9511	0.9660	0.9297	0.9520	0.0703	0.0336	0.9590	0.9843

On the test set, both stacked models preserved high accuracy and sensitivity, see Table 5. The *Stacked Random Forest* and *Stacked GLM* showed the best **Sensitivity** (0.974 and 0.960) and **F1 Scores** (0.930 and 0.926). However, both ensemble models had **high False Positive Rates** (FPR = 0.637 and 0.598), resulting in **lower Specificity** (0.363 and 0.402). Traditional models like *KNN* and *RF* demonstrated high Specificity (>0.98) but very low Sensitivity (<0.23), failing to identify many true positive cases. Ensemble models had the high AUC values (*Stacked GLM*: 0.835, *Stacked RF*: 0.814), indicating strong overall classification ability.

Table 5: Model Performance Comparison in the Test Set

Model	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR	F1 Score	AUC
GLM	0.8546	0.4938	0.9229	0.5479	0.0771	0.5062	0.5195	0.8103
BAGGED	0.8566	0.3210	0.9579	0.5909	0.0421	0.6790	0.4160	0.8176
KNN	0.8585	0.1605	0.9907	0.7647	0.0093	0.8395	0.2653	0.6901
RF	0.8605	0.2222	0.9813	0.6923	0.0187	0.7778	0.3364	0.8168
GBM	0.8625	0.3580	0.9579	0.6170	0.0421	0.6420	0.4531	0.8310
Stacked GLM	0.8711	0.9600	0.4017	0.8953	0.5983	0.0400	0.9260	0.8350
XGB	0.8762	0.3704	0.9720	0.7143	0.0280	0.6296	0.4878	0.8452
Stacked RF	0.8768	0.9740	0.3628	0.8907	0.6372	0.0260	0.9301	0.8140

In summary, for the above results, both ensemble models demonstrated substantial gains in recall (sensitivity) and F1 score, indicating superior performance in identifying true positive (i.e., fatal) cases. While traditional models such as `GLM` or `kNN` exhibited high specificity, they performed poorly in identifying deaths, making them less suitable for high-stakes applications where missing fatal cases is costly. The trade-off in ensemble models was a reduction in specificity, suggesting a higher rate of false positives. This is expected in models trained using SMOTE and tuned for sensitivity but highlights the need for calibration and possibly cost-sensitive learning.

Discussion

This study set out to predict in-hospital mortality following myocardial infarction by applying a stacked ensemble machine learning framework to a high-dimensional, imbalanced clinical dataset. Our results support the hypothesis that ensemble methods, particularly stacking, can enhance model performance over individual classifiers in the context of imbalanced health data. Both ensemble models achieved high sensitivity (>0.95) and F1 scores (>0.92) on the test data, highlighting their ability to correctly identify patients at risk of early death. These gains are particularly relevant in public health settings, where under-identification of high-risk individuals can have serious consequences. In contrast, individual models such as `logistic regression` and `kNN` achieved high specificity but very low sensitivity, rendering them less useful in identifying fatal complications. One major insight is that *stacked models* integrated the strengths of individual classifiers to capture complex relationships in the data. However, there was **sensitivity-specificity trade-off** introduced by oversampling and ensemble techniques. While the stacked models were effective in flagging deaths, they also increased false positives. This highlights the need for **calibration curves**, **cost-sensitive metrics**, and **clinical validation** to evaluate their real-world utility. Limitations of this study include the reliance on a single dataset without external validation, potential bias from synthetic sampling (SMOTE) that may have introduced synthetic patterns that reduce model generalizability, and the absence of uncertainty quantification in model predictions. The model was also optimized for binary classification of one outcome (`LET_IS`) despite the availability of multi-complication outcomes. Lastly, the high **False Positive Rates** in stacked models could lead to over-diagnosis or unnecessary follow-ups.

Future work should aim to enhance the generalizability and interpretability of the proposed ensemble framework. **External validation** using more recent and diverse clinical datasets is crucial to assess model performance across different populations and care settings. To ensure clinical reliability, **calibration assessments** such as calibration plots or the Brier score should be used to evaluate the accuracy of predicted probabilities. Expanding the model to support **multitask learning** could allow simultaneous prediction of all 12 complication outcomes, improving efficiency and capturing shared risk patterns. Additionally, incorporating **SHapley Additive exPlanations (SHAP)** or permutation-based importance metrics would enhance interpretability, particularly within the stacked ensemble. Lastly, replacing simple imputation with **multiple imputation** methods may better address missing data while preserving data integrity for modeling.

Data Availabiliy

All codes and datasets for this project can be found at my GitHub Repository, which also has the link to Project Website.

References

1. Roth, G. A., Mensah, G. A., Johnson, C. O., Addolorato, G., Ammirati, E., Baddour, L. M., Barengo, N. C., Beaton, A. Z., Benjamin, E. J., Benziger, C. P., et al. (2020). Global burden of cardiovascular diseases and risk factors, 1990–2019: Update from the GBD 2019 study. *Journal of the American College of Cardiology*, *76*(25), 2982–3021.
2. James, S. L., Abate, D., Abate, K. H., Abay, S. M., Abbafati, C., Abbasi, N., Abbastabar, H., Abd-Allah, F., Abdela, J., Abdelalim, A., et al. (2018). Global, regional, and national incidence, prevalence, and years lived with disability for 354 diseases and injuries for 195 countries and territories, 1990–2017: A systematic analysis for the global burden of disease study 2017. *The Lancet*, *392*(10159), 1789–1858.
3. Groenewegen, A., Rutten, F. H., Mosterd, A., & Hoes, A. W. (2020). Epidemiology of heart failure. *European Journal of Heart Failure*, *22*(8), 1342–1356.
4. Anderson, J. L., Adams, C. D., Antman, E. M., Bridges, C. R., Califf, R. M., Casey, D. E., Chavey, W. E., Fesmire, F. M., Hochman, J. S., Levin, T. N., et al. (2013). 2012 ACCF/AHA focused update incorporated into the ACCF/AHA 2007 guidelines for the management of patients with unstable angina/non–ST-elevation myocardial infarction: A report of the american college of cardiology foundation/american heart association task force on practice guidelines. *Journal of the American College of Cardiology*, *61*(23), e179–e347.
5. Kutty, R. S., Jones, N., & Moorjani, N. (2013). Mechanical complications of acute myocardial infarction. *Cardiology Clinics*, *31*(4), 519–531.
6. Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer.
7. Dietterich, T. G. (2000). Ensemble methods in machine learning. *International Workshop on Multiple Classifier Systems*, 1–15.
8. Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259.
9. Abbas, S., Ojo, S., Krichen, M., Alamro, M. A., Mihoub, A., & Vilcekova, L. (2024). A novel deep learning approach for myocardial infarction detection and multi-label classification. *IEEE Access*.
10. Diakou, I., Iliopoulos, E., Papakonstantinou, E., Dragoumani, K., Yapijakis, C., Iliopoulos, C., Spandidos, D. A., Chrousos, G. P., Eliopoulos, E., & Vlachakis, D. (2024). Multi-label classification of biomedical data. *Medicine International*, *4*(6), 68.
11. Satty, A., Salih, M. M., Hassaballa, A. A., Gumma, E. A., Abdallah, A., & Khamis, G. S. M. (2024). Comparative analysis of machine learning algorithms for investigating myocardial infarction complications. *Engineering, Technology & Applied Science Research*, *14*(1), 12775–12779.
12. Ghafari, R., Azar, A. S., Ghafari, A., Aghdam, F. M., Valizadeh, M., Khalili, N., & Hatamkhani, S. (2023). Prediction of the fatal acute complications of myocardial infarction via machine learning algorithms. *The Journal of Tehran University Heart Center*, *18*(4), 278.
13. Tarabanis, C., Kalampokis, E., Khalil, M., Alviar, C. L., Chinitz, L. A., & Jankelson, L. (2023). Explainable SHAP-XGBoost models for in-hospital mortality after myocardial infarction. *Cardiovascular Digital Health Journal*, *4*(4), 126–132.